Fall 11-26-2013

Time Resloved SAXS Analysis of Fusion Protein

Matthew Andorf
DePaul University, matthewbandorf@lewisu.edu

TIME RESOLVED SAXS ANALYSIS OF FUSION PROTIEN

A Thesis

Presented in

Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

August, 2 0 1 3

BY

Matthew Andorf

PHYSICS DEPARTMENT

College of Science and Health

DePaul University

Chicago, Illinois

Acknowledgements

Dr. Landahl

Dr. Sarma

Dr. Pando

Taylor Poor (Northwestern University)

Srinivals Chakravarthy (Advanced Photon Source Bio-CAT Sector 18)

....and all the friends and family that helped along the way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES – *Continued*

ABSTRACT

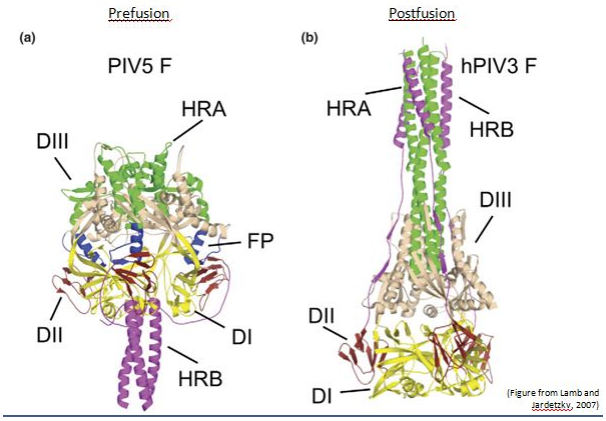
The structural dynamics of the fusion protein (F protein) of parainfluenza virus 5 (PIV 5) was studied using time resolved Small Angle X-ray Scattering (SAXS). Conformational changes of the F protein, equivalent to the conformational changes the protein must undergo while infecting a host cell, are triggered by temperature jump to the protein in solution. The time and temperature dependence of the rate of the protein changing state is then determined through standard SAXS analysis. Thermodynamic transition state theory is then used to calculate the activation energy, enthalpy and entropy of the transition which is independent of any particular model. It is determined that the transition is entropically driven. Additionally a new technique employing constrained singular value decomposition is used to decompose the SAXS scattering patterns into basis states. The new technique successfully corroborates results found using the standard SAXS analysis.

CHAPTER 1

Introduction

This thesis presents the first study of the structural dynamics of the fusion protein of parainfluenza virus 5. Parainfluenza virus 5 (PIV5) is a member of the paramyxovirus family of enveloped viruses and relies on a fusion protein (F) and an attachment protein, hemaglutinin-neuraminidase (HN), to effect membrane fusion and gain entrance into host cells (Sec. 1.2). Like the fusion proteins of other enveloped viruses such as influenza and HIV, the F protein of PIV5 is translated in a metastable state (pre-fusion), ready for the stimulus from HN that will trigger its activation and refolding into a post-fusion form. While crystal structures exist for the pre- and post-fusion forms of F (PIV5 and hPIV3, respectively), very little is understood about how this metastable protein accomplishes its transformation from pre- to postfusion, effecting fusion of the viral envelope with the host cell membrane in the process. PIV5 F is a Class I fusion protein, and as such serves as a model system for understanding the fusion proteins and entry mechanisms of other Class I viruses, including such medically important pathogens as influenza, human immunodeficiency virus (HIV), Newcastle disease virus (NDV), measles virus, and Ebola.

This thesis is motivated by the ultimate goal of determining the structural basis for the conformational changes that occur during the fusion reaction of the paramyxovirus PIV5 F protein. As it is one of the primary models for the fusion reactions of other class I viruses, a deeper understanding of the fusion mechanism of PIV5 fusion could help guide the analysis of other viral fusion systems and the development of drugs that interfere with them.



h

Figure 1.1: The pre and post fusion structures of the fusion protein of parainfluenza virus 5. (Yin et al. (2006))

1.1 Conformational Change

Large molecules, such as proteins, undergo conformational changes governed by the thermodynamics of its environment. A conformational change results in large scale structural (at the molecular level) changes as well as changes in the proteins energy state.

Biology makes use conformational changes by the fact that a protein may be active in one conformational state and inactive in another. Specifically for F, conformational changes are used to allow a virus to enter its genetic material into a host cell (Sec. 1.2).

1.2 Previous study

Previous studies of PIV5 F suggest the following model of viral entry into the host cell (Yin et al. (2006)). The HRB Helix seen on the bottom of the prefusion state (Fig 1.1) melts while keeping the rest of the prefusion state intact. Initiated by refolding of DIII, the HRA helices then unravels into a hairpin like structure and binds to the host cell membrane. With leverage provided from the HRA and HRB

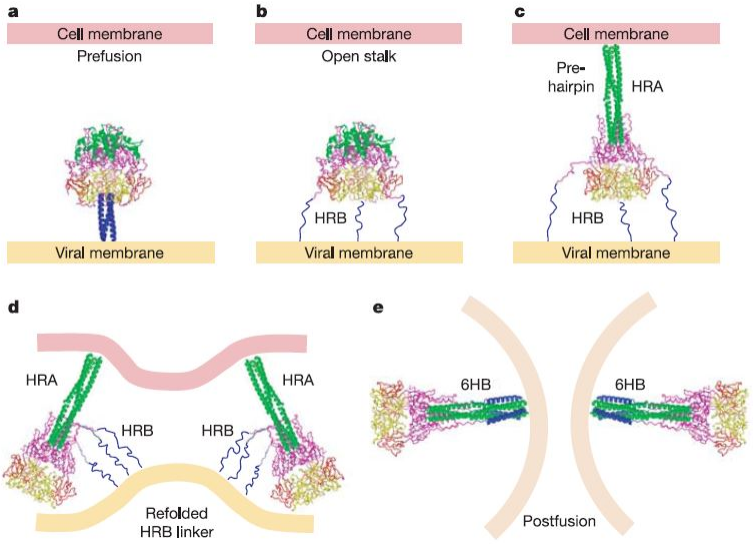


Figure 1.2: A proposed model of viral entry into a host cell based on previous work. (Yin et al. (2006))

helices, F is able to torque viral and host cell membranes together, and bind them into one. The post-fusion structure forms inside the host cell and the process is complete. This process is visualized in Fig. 1.2. This fusing process is what gives the F protein its name.

The model presented is based on the solved crystal structures of PIV5 F pre- and post- fusion states, however it is not known if any intermediate states exist. As an indepedent confirmation of the crystal structure, static Small Angle X-Ray Scattering (SAXS) measurements of a temperature stabilized version of PIV5 F are able to recapitulate the structure remarkably well (Fig. 1.3). Additionally since SAXS experiments take place with the protein in solution, which is nearer to physiological condition, the transition pathway between the pre and post fusion structures can possibly be observed.

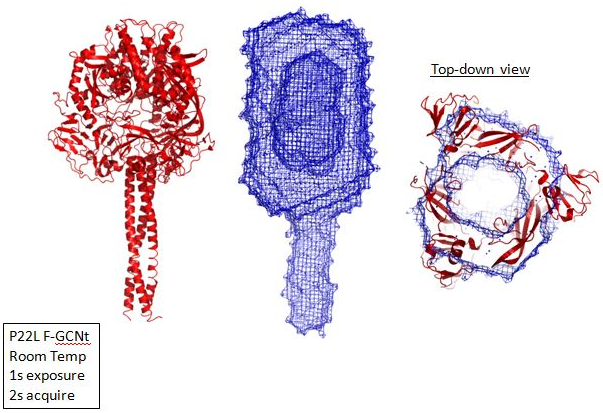


Figure 1.3: A comparison of the crystal structure(left) to the reconstructed structure with SAXS data (middle). The extreme right image is a top down view of the crystal structure and SAXS reconstruction overlayed.

1.3 Approach

The structural dynamics are probed using SAXS, done at the Advanced Photon Source (APS) at Argonne National Laboratory. A sample of the prefusion protein is subjected to a sudden rise in temperature (a temperature jump). Ordinarily the triggering activation for the protein occurs with binding to HN. However Connolly and colleagues have shown using biological activity and static image electron microscopy, that heat can be used as a trigger in the absence of an attachment protein and that heat induced conformation of the F protein resembles the assumed prefusion F when triggered physiologically (Connolly et al. (2006)). Therefore observing the F protein after a temperature jump is equivalent to the process of viral entry in terms of protein conformational changes.

The experimental procedure and analysis is outlined here, for a more detailed discussion see Chapters 4 and 5. The scattering pattern of the sample is observed at various time increments. Then a Guinier analysis (Sec. 5.2) is performed allowing the radius of gyration, R_G to be observed as a function of time. By jumping to several different final temperatures (Sec. 4.2), the rate of population change can be

measured as a function of temperature. This then can used to find thermodynamic quantities such as the activation energy, enthalpy and entropy of reaction (Sec. 5.3) .

CHAPTER 2

SAXS

The structural dynamics of the fusion protein of PIV 5 was studied using Small Angle X-Ray Scattering (SAXS). SAXS utilizes elastic collisions between X-ray photons and electrons to record a scattering pattern. This technique is advantageous over crystallography in that the protein is in a fluid and thus allowed to change form, which is nearer to physiological conditions. However this advantage does come at the expense of spatial resolution. In this section I will present the underlying theory to SAXS analysis, namely the creation and interpretation of the scattering patterns formed in experiments.

2.1 Experimental set up

The experiment took place at the Advanced Photon Source (APS) at Argonne National Laboratory. The APS is a 3rd generation synchrotron radiation facility that provides an well collimated and extremely high average brightness monochromatic X-Ray beam.

The set up of a SAXS experiment is relatively straight forward (Fig. 2.1). The X-Ray beam passes through apertures to control its divergence and then passes through the sample [1]. The sample scatters the light and a position sensitive detector collects photon counts, which are turned into intensities. Since the sample is in a solution, each run of the experiment is performed with just the solution and then with sample and solution, which allows for background subtraction.

The detector collects two dimensions of data, radial and azimuthal. Since the

[1] For a more detailed explanation of the creation and refining of the beam see section (4.1)

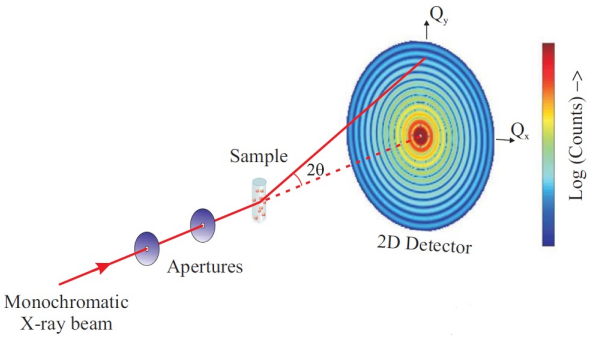


Figure 2.1: The SAXS experimental set up. Not shown here is a stop-flow pump and an apparatus to induce a temperature jump(Nielsen and McMorrow (2011)).

scattering pattern is the result of an ensemble of proteins which are randomly oriented with equal probability, the signal is isotropic and so the azimuthal components are added, giving a single radial profile. In order to analyse the data scattering events are assumed independent. This condition is met by keeping the solution rather dilute ($\sim$ 1 mg/ml).

2.2 Scattering from two electrons

To understand the scattering pattern formed by a protein in solution we will follow the approach of (Nielsen and McMorrow (2011)) and start with the most simple system possible, two electrons separated by a distance R. One of the electrons is placed at the origin. Now consider what happens when both electrons are illuminated by a beam described by the vector, $2\frac{\pi}{\lambda}\hat{k} = \mathbf{k}$. At both electrons the beam will deflect in the $\mathbf{k}'$ direction, however the path length of a photon is different depending on which electron the photon interacted with. Therefore a phase difference between two photons, initially in phase, will occur. This phase difference is equal to:

$$\varphi = 2\pi\frac{\Delta P}{\lambda} \tag{2.1}$$

where ΔP is the path length difference between the photons. The path length

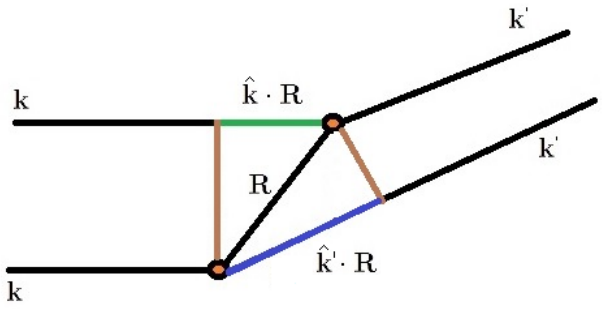


Figure 2.2: The Geometry of the two electron system. The green segment is $\hat{k} \cdot \mathbf{R}$ and the blue segment is $\hat{k'} \cdot \mathbf{R}$. The tan segments are drawn in to help illustrate the geometry.

difference is found by considering the projection of $\mathbf{k}$ and $\mathbf{k}'$ on to the position vector $\mathbf{R}$ (Fig. 2.2). Note that since the scattering is elastic the magnitude of $\mathbf{k}$ and $\mathbf{k}'$ are equal.

$$\Delta P = \hat{k} \cdot \mathbf{R} - \hat{k'} \cdot \mathbf{R} = \mathbf{R} \cdot (\hat{k} - \hat{k'}) \tag{2.2}$$

We define the quantity $\mathbf{Q}$ as the scattering vector,

$$\mathbf{Q} \equiv \mathbf{k} - \mathbf{k}' \tag{2.3}$$

and thus

$$\varphi = \mathbf{R} \cdot (\mathbf{k} - \mathbf{k}') = \mathbf{R} \cdot \mathbf{Q} \tag{2.4}$$

If we define the angle between $\mathbf{k}$ and $\mathbf{k}'$ as 2θ then the magnitude of Q is given by

$$|\mathbf{Q}| = 2ksin(\theta) = 4\frac{\pi}{\lambda}sin(\theta). \tag{2.5}$$

The amplitude of two objects with a scattering amplitudes A_1 and A_2 can be written as

$$A(\mathbf{Q}) = A_1 + A_2 e^{i\varphi}$$

and the intensity is given by

$$I(\mathbf{Q}) = A_1^2 + A_2^2 + A_1 A_2 e^{i\mathbf{R}\cdot\mathbf{Q}} + A_1 A_2 e^{-i\mathbf{R}\cdot\mathbf{Q}}.$$

Consider what would happen if the two electron system was allowed to orient randomly with respect to the beam and the intensities of these random orientations were averaged, then the intensity becomes

$$\langle I(\mathbf{Q})\rangle_{Orient.av} = A_1^2 + A_2^2 + 2A_1A_2\langle e^{i\mathbf{R}\cdot\mathbf{Q}}\rangle_{Orient.av} \tag{2.6}$$

with

$$\langle e^{i\mathbf{R}\cdot\mathbf{Q}}\rangle_{Orient.av} = \frac{sin(QR)}{QR}$$

As long as scattering events remain independent, the concepts found in studying the two electron system readily carry over to a system of N electrons of the same scattering amplitude A_0. For such a system the amplitude is written as

$$A(\mathbf{Q}) = A_0 \sum e^{i\mathbf{r_j}\cdot\mathbf{Q}} \tag{2.7}$$

and the intensity is then found by taking the modulus

$$I(\mathbf{Q}) = |A_0 \sum e^{i\mathbf{r_j}\cdot\mathbf{Q}}|^2. \tag{2.8}$$

2.3 Scattering from more than two electrons

The next step in complexity is scattering from a single atom. The atom will have a classical charge distribution, $\rho(\mathbf{r})$. At this point it is convenient to introduce the form factor

$$f^0(Q) = \int \rho(\mathbf{r})e^{iQ\cdot\mathbf{R}}dv \tag{2.9}$$

which can be thought of as the contribution to scattering from a volume element dv at the position $\mathbf{r}$ with a certain phase factor. The form factor can also be recognized as a Fourier Transform. It is in this way that structural information of the system is obtained through analysing its diffraction pattern.

From the superposition of atoms, the molecular version of the atomic form can be written as

$$F^{mol} = \sum_j f_{j,atomic}(\mathbf{Q})e^{iQ\mathbf{r_j}} \tag{2.10}$$

where j is for the jth atom in the molecule. The molecular form factor then is the discrete sum of scattering contributions of individual atoms, which are found through integration of Eq.2.9. If the molecule is in solution then Eq.2.10 can be summed over each molecule. By taking the modulus of such a sum the intensity can be written as[2]

$$I(\mathbf{Q}) = f(\mathbf{Q})^2 \sum_n e^{iQ\mathbf{r_n}} \sum_m e^{-iQ\mathbf{r_m}} \tag{2.11}$$

By recognizing the terms $m = n$ can be written as $Nf(\mathbf{Q})^2$, letting the sum over m turn into an integral and adding and subtracting a term proportional to ρ_{av} the above equation can be rewritten as

$$I(\mathbf{Q}) = Nf(\mathbf{Q})^2 + f(\mathbf{Q})^2 \sum_n \int (\rho_n(\mathbf{r}_{nm}) - \rho_{av)} e^{i\mathbf{Q}\cdot(\mathbf{r_n}-\mathbf{r_m})} dv_m - f(\mathbf{Q})^2 \sum_n \int \rho_{av} e^{i\mathbf{Q}\cdot(\mathbf{r_n}-\mathbf{r_m})} dv_m \tag{2.12}$$

Small Q corresponds to large real space. For large real space ρ_{av} approaches ρ_n and therefore the second term in the equation approaches zero. Furthermore since $Q \propto \sin\theta$, small Q corresponds to small angles. Therefore the third term in the above equation is the intensity at small angles, $I(\mathbf{Q})^{SAXS}$. By turning the sum over n into an integral and defining $\rho_{av} f \equiv \rho_{sl}$, The SAXS intensity can be written as

$$I(\mathbf{Q})^{SAXS} = |\int \rho_{sl} e^{i\mathbf{Q}\cdot(\mathbf{r})} dv|^2 \tag{2.13}$$

2.4 Determining the Radius of Gyration with SAXS

A strength in SAXS analysis is the ability to determine large scale structure and morphology of a particle. In this section we consider the scattering pattern formed by identical particles in a dilute solution, in this case the (Guinier (1994)) $I(\mathbf{Q})^{SAXS}$ becomes

$$I(\mathbf{Q})^{SAXS} = (\rho - \rho_0)^2 |\int e^{i\mathbf{Q}\cdot(\mathbf{r})} dv|^2 = (\rho - \rho_0)^2 f(\mathbf{Q})^2 \tag{2.14}$$

[2]To avoid cluttered notation from this point on, unless otherwise stated, $f(\mathbf{Q})$ is the molecular form factor

where ρ is the electronic density of the particles in the solution of electronic density ρ_0 and $f(\mathbf{Q})$ is the form factor.

For small angles $\mathbf{Q}$ is nearly perpendicular to the direction of the incident X-Ray beam, $\mathbf{k}$, we will denote this direction as $\mathbf{D}$. Then $\mathbf{r} \cdot \mathbf{Q} = Qr_D$ where r_d is the projection of $\mathbf{r}$ along $\mathbf{D}$. The form factor can now be written as

$$f(\mathbf{Q}) = \int e^{iQr_d} \sigma_{r_D} dr_D \tag{2.15}$$

here σ_{r_D} is the cross sectional area of the particle along a plane normal to D at a distance r_D. The origin is selected to be the center of mass of a particle. Since Qr_D is small, the exponential can be expanded to the second order.

$$f(\mathbf{Q}) = \int \sigma(r_D) dr_D + Q \int r_D \sigma(r_D) dr_D - \frac{Q^2}{2} \int r_D^2 \sigma(r_D) dr_D \tag{2.16}$$

The first term is zero since we have chosen the origin to be the center of mass. The integral in the second term can be recognized as the volume of the particle, V. Next we set

$$R_D^2 = \frac{1}{V} \int r_D^2 \sigma(r_D) dr_D \tag{2.17}$$

which is the average inertial distance along $\mathbf{D}$. Now $f(\mathbf{Q})$ can be written as

$$f(\mathbf{Q}) = V - Q^2 V R_D^2 \approx V e^{\frac{-Q^2}{2} R_D^2} \tag{2.18}$$

since

$$e^{-x} \approx 1 - x.$$

The intensity per particle may be written as

$$I(\mathbf{Q})^{SAXS} = (\rho - \rho_0)^2 V^2 e^{-Q^2 R_D^2}. \tag{2.19}$$

However for the experiment we measure the scattering pattern of many particles which are randomly oriented in solution. In this case R_D^2 in the above equation

must be replaced with $\overline{R_D^2}$, the average. The average can be found by using two more Cartesian axis **U** and **V**. In which case

$$R_G^2 = R_D^2 + R_V^2 + R_U^2 = \frac{\int r^2 dv}{V} \tag{2.20}$$

with R_G^2 being the radius of gyration and R_U and R_V being the inertial distances to their respective planes. As the particle rotates around the origin, R_G^2 remains constant. The assignment of **D** was arbitrary and so on the average R_D,R_U and R_V are equal, thus $3\overline{R_D^2} = R_G^2$. The scattering intensity can now be written as

$$I(\mathbf{Q})^{SAXS} = (\rho - \rho_0)^2 V^2 e^{\frac{-Q^2 R_G^2}{3}}. \tag{2.21}$$

Recalling that $|Q| = \frac{4\pi}{\lambda} sin(\theta)$ The intensity may be written in terms of the scattering angle (2θ)

$$I(q) = (\rho - \rho_0)^2 V^2 e^{\frac{16\pi^2 R_G^2 \theta^2}{3\lambda^2}}. \tag{2.22}$$

in the above, the small angle approximation of the sine function was used. Notice (2.21) and (2.22) do not require knowledge of the phase, this is important because the phase can not be measured. Intensity in this form also allows us to determine the R_G by plotting $log(I)$ vs Q^2.

$$log(I) = log((\rho - \rho_0)^2 V^2) - \frac{Q^2 R_G^2}{3} \tag{2.23}$$

and performing a linear curve fit. Then R_G is given by

$$R_G = \sqrt{-3m} \tag{2.24}$$

where m is the slope. Such a plot is called a Guinier plot. Although we can determine R_G this way we can not determine anything about the shape since for any particular R_G there are an infinite amount of shapes.

2.5 Heterogeneous particles

The above derivation assumes the particle is homogeneous (that is, it's electron density was constant in the particle); however, the results found in the previous

section can be generalized to a particle with an arbitrary electron distribution. In section 2.2 we derived the average intensity of an ensemble of randomly oriented electrons. A similar analysis can be done with a particle and leads to

$$I_N(Q) = \sum_{k=1}^{n}\sum_{j=1}^{n} f_k f_j \frac{sin(Qr_{kj})}{Qr_{kj}} \tag{2.25}$$

here f is the atomic form factor and the indices are for the kth and jth atom in the particle. This equation is known as the Debye formula.The sine function can be expanded using

$$sin(x) \approx x - \frac{x^3}{3!}$$

So the Debye equation (Eq. 2.25) becomes

$$I_N(Q) = \sum_{k=1}^{n}\sum_{j=1}^{n} f_k f_j - \sum_{k=1}^{n}\sum_{j=1}^{n} f_k f_j \frac{Q^2 r_{kj}^2}{6} \tag{2.26}$$

The origin is selected such that $\sum f_k r_k = 0$. r_{kj} can be written as

$$r_{kj}^2 = (\mathbf{r_j} - \mathbf{r}_k)^2 = r_k^2 - r_j^2 - 2\mathbf{r_k} \cdot \mathbf{r_j} \tag{2.27}$$

multiply by $\sum f_j \sum f_k$ to get

$$\sum f_j \sum f_k r_{kj}^2 = \sum f_j \sum f_k r_k^2 - \sum f_j \sum f_k r_j^2 - \sum f_j \sum f_k 2\mathbf{r_k} \cdot \mathbf{r_j}. \tag{2.28}$$

The first two terms can be factored to give

$$\sum f_j \sum f_k r_k^2 = n \sum f_k r_k^2 \tag{2.29}$$

the third term is zero because of our coordinate choice and n is the number of electrons in the particle. We set

$$R_H^2 = \frac{\sum f_k r_k^2}{\sum f_k} \tag{2.30}$$

here the H is used to denote heterogeneous. Next by substituting equation (2.29) into equation (2.26) and multiplying the top and bottom of the second term by $\sum f_k$, we get back

$$I_n(Q) = n^2(1 - \frac{Q^2 R_H^2}{3}) \approx \rho^2 V^2 e^{\frac{Q^2 R_H^2}{3}} \tag{2.31}$$

where we used $n = \rho v$. Again the approximation $e^{-x} \approx 1 - x$ was used. Equation (2.31) is in agreement with (2.21) and therefore the results found in section (2.4) are valid for a particle of arbitrary electron density.

2.6 Non-identical particles

In our experiment, we our observing the diffraction pattern of a solution which contains a protein in more than one state. If the solution is dilute enough to keep scattering events independent the scattering pattern is the superposition of each different protein state. Thus for a solution containing m different states Eq. 2.21 becomes

$$I(\mathbf{Q}) = \sum_{k=1}^{m} p_k n_k^2 e^{\frac{-Q^2 R_k^2}{3}} \tag{2.32}$$

where p_k is the relative population of the kth state, n_k is the number of electrons in the state, and R_k is its respective radius of gyration. Next if we expand out Eq. (2.32) using $(e^{-x} \approx 1 - x)$ we can get

$$I((Q)) = \sum_{k=1}^{m} p_k n_k^2 \left(1 - \frac{\sum_{k=1}^{m} p_k n_k^2 Q^2 R_k^2}{3 \sum_{k=1}^{m} p_k n_k^2} \right) \tag{2.33}$$

We see a plot of log(I) vs Q^2 will have a slope of $-3R_{non-id}$ with R_{non-id} defined as

$$R_{non-id}^2 = \frac{\sum p_k n_k^2 R_k^2}{\sum p_k n_k^2} \tag{2.34}$$

For a protein changing state, so long as it does not bind to another protein or break apart, the number of electrons on it will stay the same. In this case the n_k in Eq. (2.34) is common to all terms and so cancels. Furthermore if protein is only observed in two states, the above equation simplifies to

$$R_{non-id}^2 = p_1 R_1^2 + p_2 R_2^2. \tag{2.35}$$

Notice that since the radii terms in Eq.(2.35) are squared the value of R_{non-id} is determined more by the values of R_1 and R_2 then by their respective populations. Because the populations must add to one we have

$$p_1 + p_2 = 1 \tag{2.36}$$

In this experiment p is observed to be both a function of time and temperature. In the next section the thermodynamic causes and implications of this will be developed.

CHAPTER 3

THERMODYNAMICS

3.1 The Boltzmann distribution

The previous section mentioned that the relative population is a function of both time and temperature. To see why let us first consider a collection of particles that can occupy different energy states, E_k. In regards to a protein these different energy states may be different conformational states. When a protein undergoes a conformational change, there may be a change in both its enthalpy and entropy.

The enthalpic change is the result interactions between amino acids via van der Waal and electrostatic forces, as well as hydrogen bond breaking or forming. Additionally amino acids once buried in a protein may now be exposed to the solution resulting in different energy interactions. The entropic change results from a change in elasticity (Middleton et al. (2001)). Simply put the more elastic the protein becomes, the more microstates it can form with the same energy, or the same macrostate.

A well known result of statistical mechanics is that an ensemble of particles of different energy states is governed by the Boltzmann distribution (Dill (2011))

$$p_k = \sum \frac{e^{\frac{-E_k}{RT}}}{Z} \tag{3.1}$$

where T is the temperature R is the ideal gas constant and Z is the partition function given by

$$Z = \sum W(E_k) e^{\frac{-E_k}{RT}}. \tag{3.2}$$

W is the multiplicity of a particular state (i.e. a degenerate state has $W > 1$). The

relative population of two states,p_k and p_j may be written as

$$\frac{p_k}{p_j} = \frac{W(E_k)}{W(E_j)} e^{\frac{(-E_k - E_j)}{RT}} \tag{3.3}$$

The Boltzmann distribution assumes thermodynamic equilibrium. Suppose a system is in thermodynamic equilibrium such that its population is described by the Boltzmann distribution when suddenly the temperature jumps to a new constant temperature T_h. How might the populations move towards the new equilibrium?

3.2 Transition State Theory

Consider a change of state such as

$$A \xrightarrow{\kappa} P \tag{3.4}$$

that proceeds at an observed rate κ. If in order for the above to happen, A must first transform into an intermediate unstable state T then this reaction may be written as

$$A \xrightarrow{K^{\ddagger}} T \xrightarrow{\kappa^{\ddagger}} P \tag{3.5}$$

with $K^{\ddagger}$ being the equilibrium constant between A and T, and $\kappa^{\ddagger}$ being the rate coefficient between T and P. Since the transition state is unstable the time scale of its existence is usually too short to be observed, with the exception being in ultra fast time resolved experiments. Therefore in this experiment $K^{\ddagger}$ and $\kappa^{\ddagger}$ are not observed.

Any equilibrium constant can be expressed as a ratio of each species present at equilibrium, $K^{\ddagger} = N_T/N_A$. The process we are interested in occurs at a fixed temperature and pressure. Therefore it is the Gibb's free energy that defines equilibrium for our system

$$dG = -SdT + PdV + \mu_A dN_A + \mu_T dN_T = 0 \tag{3.6}$$

with μ being the chemical potential. However the above simplifies because of the constant pressure and temperature. Furthermore since the number of total particles must stay constant we have

$$N_A + N_T = N_{Total} \Longrightarrow dN_A = -dN_T$$

Then equation (5.2) becomes

$$(\mu_A - \mu_T)dN_A = 0$$

The above must hold true for any dN_A and thus

$$\mu_A = \mu_T \tag{3.7}$$

In general the chemical potential of a species can be expressed as

$$\mu = -\varepsilon_0 ln\left(\frac{z}{N}\right) = RTln\left(\frac{ze^{\frac{-\varepsilon_0}{RT}}}{N}\right) \tag{3.8}$$

with ε_0 being the ground state of the state and z being the partition function of the particular state. Combining equations (3.7) and (3.8) gives

$$\frac{z_A e^{\frac{-\varepsilon_{0A}}{RT}}}{N_A} = \frac{z_T e^{-\frac{\varepsilon_{0T}}{RT}}}{N_T} \tag{3.9}$$

and recalling how the equilibrium constant is defined, $K^{+}_{+} = \frac{N_T}{N_A}$, we get

$$K^{+}_{+} = \frac{N_T}{N_A} = \left(\frac{z_T}{z_A}\right) e^{-(\frac{\varepsilon_{0T} - \varepsilon_{0A}}{RT})} \tag{3.10}$$

Notice in the above equation K^{+}_{+} is not a true equilibrium constant since T is an unstable state. A key assumption in transition state theory is K^{+}_{+} can be treated as any true equilibrium constant.

The rate at which A turns into P can be expressed as

$$\frac{dN_P}{dt} = K^{+}_{+}T = K^{+}_{+}\kappa^{+}_{+}A \tag{3.11}$$

from this we see that the rate coefficient, κ is given by

$$\kappa = \overset{+}{K^{+}}\overset{+}{\kappa^{+}} \tag{3.12}$$

Next let's examine the partition function of the intermediate state, z_T in more detail. z_t differs from a normal partition function in that it has energy contributions from both stable degrees of freedom, such as translational and rotational motion and a nonstable degree of freedom, namely nonequilibrium vibrational energy, ν_ξ. If these degrees of freedom are independent from each other, z_t can be factored such that

$$z_t = \overset{+}{z^{+}} z^{\xi} \tag{3.13}$$

where $\overset{+}{z^{+}}$ represents the partition function of the stable degrees of freedom and z^{ξ} represents the vibrational degrees of freedom. The vibrational degrees of freedom result from the fact that there must be some internal motion present in order for the state to change. z^{ξ} may be written as

$$z^{\xi} = \sum_{\nu=0}^{\infty} e^{-\frac{\nu h \nu_\xi}{RT}} \tag{3.14}$$

The bonds in state T are presumed weak since the state is nonstable. In this case the vibrational frequency must be small (a weak bond is akin to a low spring constant) and

$$\frac{h\nu_\xi}{RT} \ll 1 \tag{3.15}$$

This allows us to approximate equation (3.14) as

$$z^{\xi} \approx \frac{h\nu_\xi}{RT} \tag{3.16}$$

Once T is formed it will rapidly transform into P at a rate of ν_ξ and so

$$\overset{+}{\kappa^{+}} = \nu_\xi \tag{3.17}$$

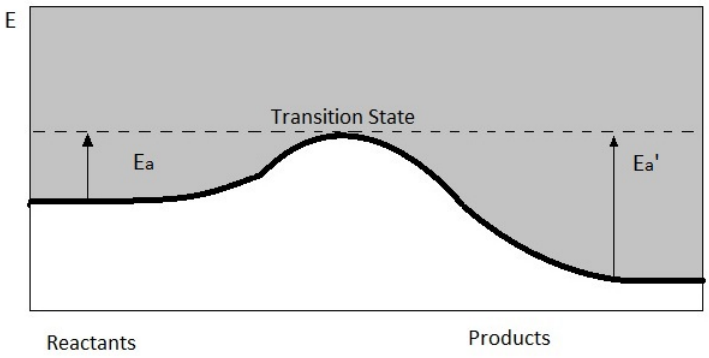


Figure 3.1: In the diagram the activation energy is shown to be the difference in energy of state A (the reactants) and the transition state.

We attain an expression of the observed rate, κ by combining equations (3.10), (3.13), (3.16) and (3.17) into equation (3.12)

$$\kappa = (\frac{RT}{h})(\frac{\overline{z^{+}}}{z_A}) = (\frac{RT}{h})\overline{K^{+}} \tag{3.18}$$

the over bar indicates the nonstable degree of freedom has been factored out. Finally using $RTln(\overline{K^{+}}) = \Delta G$, κ can be written as

$$\kappa = (\frac{k_b T}{h})e^{\frac{-\Delta G}{RT}} = (\frac{k_b T}{h})e^{\frac{-\Delta H}{RT}}e^{\frac{\Delta S}{R}} \tag{3.19}$$

The above equation is known as the Eyring equation. By doing a linear curve fit on $ln(\frac{\kappa}{T})$ vs $\frac{1}{T}$ the enthalpy and entropy of the change of state can be found as

$$\Delta H = -m_e R \tag{3.20}$$

$$\Delta S = R(b_e - ln(\frac{R}{h})) \tag{3.21}$$

respectively, with m_e being the slope of the linearized Eyring equation and b_e being the intercept.

The Eyring equation has a very similar form to an empirical equation often used in chemistry, the Ahhrenius Equation

$$\kappa = Ae^{\frac{-E_A}{RT}} \tag{3.22}$$

A is a constant, and E_A is the activation energy. The activation energy is the difference in energy between some state A and state T (Fig.2.1). Then by comparing the Ahhrenius equation to the Boltzmann distribution it is seen that the rate of change is simply proportional to the relative population of particles that have attained enough energy to be "activated". A plot of $log(\kappa)$ vs $1/T$ will be linear with the slope, m_a , equal to $\frac{E_A}{R}$ or

$$E_A = -Rm_A \tag{3.23}$$

it is in this way the activation energy of a change of state can be found.

CHAPTER 4

The experiment

4.1 Set up

The experiment was done at Argonne National Laboratory's, Advanced Photon Source (APS), Bio-Cat Sector 18. Bio-Cat specializes in the study of the structure and dynamics of non-crystalline biological systems. The undulator at sector 18 is capable of producing a range of photon energies between 3.2-14 KeV and a relatively small unfocused beam with dimensions of $\sim 35\mu M$ by $\sim 135\mu M$ FWHM at the detector (Fischetti et al. (2004)).

The beam passes through two Silicon monochromators, then is reflected off a mirror, which is used to reject harmonics as well as vertically focus the beam, and finally passes through a collimator before impinging the sample in solution. The scattering pattern is then recorded on a detector behind the sample. A beam stop is used to prevent the detector from being directly exposed to light from the beam.

This experiment used a novel, single photon counting, two dimensional hybrid pixel array detector called PILATUS. The detector is a solid state area detector. The fast full frame read out time (6.7 ms), which is orders of magnitude faster than previous generation X-ray detectors, was essential to this experiment (Broennimann et al. (2005)).

4.2 Procedure

Immediately prior to each run the protein was spun at 90,000 RPM for 10 minutes at 4 degrees Celsius. The solutions, protein and buffer, were kept to 1 mg/ml. The sample was then loaded into a pump at room temperature inside the experiment

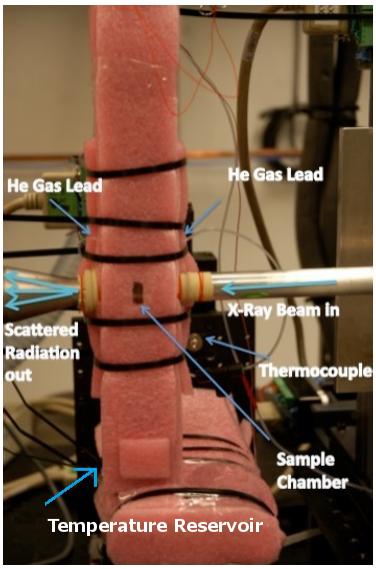


Figure 4.1: The brass mount where sample is brought to temperature and into the beam path. X-Rays move from right to left in the image.

"hutch". The hutch must be vacated before data acquisition can begin. From outside the hutch the shutters, which when closed block the X-Ray beam, are opened and a baseline reading of the beam is taken.

Meanwhile a brass fixture has been cooled or warmed to a specified temperature using circulating water from a temperature controlled reservoir. The fixture has a window in it aligned with the beam, and it is also connected to the pump via a thin plastic capillary tube. Attached to the fixture is a thermocouple which allows for real time temperature monitoring with a digital voltmeter. The pump can be controlled from outside the hutch and when activated will suck up some of the sample into the capillary. It is in this manner that sample is simultaneously brought into the beam and raised to a specific temperature.

Once a baseline is established and sample is properly loaded, data acquisition can begin. At each temperature a series of 10 exposures for a duration of 1 second were taken immediately. Each exposure is separated by a 1 second pause and therefore the acquisition time of a single exposure is 2 seconds. The shutter is closed but the

Temperature (C^0)	21.5	24.9	29.1	33.3	37.4	46.3	50.6	54.6

Table 4.1: Temperatures, after jump, in the experiment

sample is kept in the mount for 40 seconds. Then after 40 seconds has passed the data acquisition portion of the procedure is repeated for a second and third time, each round being separated by a 40 second pause. The series of 10 exposures is done over a 20 second interval. Therefore the protein is observed over a time interval of 200 seconds. This procedure is done once with sample in solution and once without it, to allow for background subtraction.

Next the temperature is changed via the reservoir and the old sample is replaced with new sample at which point the entire procedure is repeated at the new temperature. The temperatures in this experiment are given in Table 4.1.

CHAPTER 5

Data Analysis

5.1 Importing data and Removing Outliers

Once the data has been obtained, as described in the previous section, analysis begins. For this experiment almost all the analysis was done in MATLAB. The data originally came in a few large text files. Code was written to import one of these files and, sort out each particular shot and write them to their own text file. The files contain three columns each with 80 rows of data. The columns represent Q, the scattering vector, I, the intensity, and the counting statistics of the intensity which serve as the error.

The sample intensities are then subtracted from the appropriate background intensities. The background subtracted error is taken to be the sum, in quadrature, of the sample and background errors.

In a more standard static SAXS experiment a shot is often repeated 10 to 20 times. One can then plot the I vs Q data and visually determine and remove outliers. Repeating shots for this experiment was not an option for two reasons. First the very nature of a dynamic experiment prevents repeating shots in the course of one acquisition period. Second the cost of both the protein and beamtime prohibit repeating the procedure at the same temperature. For these reasons criteria, other than visual, had to be established in order to remove outliers. Two algorithms were used as described below.

The first checks for negative intensities, after background subtraction, up to some specified maximum Q value (0.0836 $\mathring{A}^{-1}$). A negative intensity occurs when the background intensity at some Q value is greater then the sample intensity at

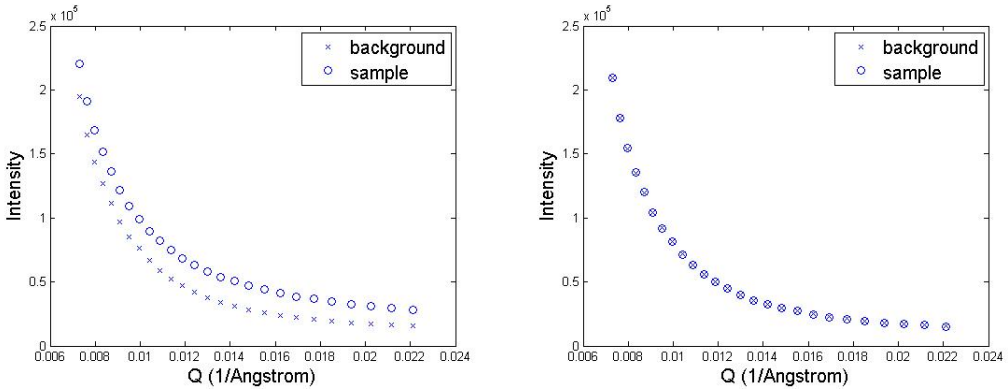


Figure 5.1: Left: A typical I vs Q plot of sample and background at 21.5 C^0 and 69 seconds, the scattering from the sample is always greater than scattering from background. Right: An example of an intensity, at 37.4 C^0 and 195 seconds that would be removed from further analysis. The sample and background are nearly identical, indicating there was a problem with loading or removing the sample.

that same Q. The maximum Q value was chosen more or less arbitrarily, but inspection of the data shows for a typical sample intensity at this Q to be $\sim 10\sigma$ above background noise. If a negative intensity is found the entire shot is removed from the analysis. The reason this algorithm is not applied to all Q values is two fold. First in general as Q increases the intensity decreases. Then at high Q the sample intensity is close to the background intensity. Second the low intensity at high Q generates poor counting statistics (compared to low Q) due to a smaller number of counts. For these two reasons background subtraction of otherwise reasonable data may result in negative intensity; and for these same reasons intensity found to be negative before 0.0836 $\mathring{A}^{-1}$ is very justified in being removed. The analysis of the data will look at Q values much lower than the maximum Q applied here, so the negative intensities past the maximum Q are inconsequential.

A negative intensity at a low Q value is most likely caused by an error in loading the sample or alternatively the sample being expelled from the sample chamber too soon. As seen in the graph on the right of (Fig. 5.1) if one of these problems occur, there is almost no difference in background and sample. Then subtraction between

these to nearly identical scattering patterns leads to negative intensities.

The next algorithm removes shots that contain an intensity that is considered too high. An unusually high intensity may be the result of an air bubble being present in the solution. The average value of all the intensities at a very low Q (where intensity was highest) were averaged. Then the maximum intensity of each shot was compared to this average. If the maximum exceeded twice the average it was considered too high and thrown out. A value that was thrown out was typically around three to four standard deviations greater than the average maximum intensity.

Lastly two vectors were made corresponding to the temperature and time of each shot. Since this information is not in the text file these vectors were created based on notes taken during the experiment.

There are numerous ways to plot intensity data aside from a simple I vs Q curve (Fig. 5.2). The most essential to this experiment is the Guinier plot. Another useful plot is the Kratky plot $I * Q^2$ vs Q. The multiplication of Q^2 to a normally decaying intensity helps emphasize the high Q (low spatial distance) portion of the scattering pattern. A Kratky plot gives information about the compactness of the protein. In general the higher the tail of a Kratky plot, the less compact the protein is.

5.2 The Guinier Analysis

In section (2) we derived an expression relating the radius of gyration, R_G to the intensity at small angles (equation 2.21). R_G can then be found from the slope of the $log(I)$ vs Q^2 plot using equation(2.24).

For a Guinier analysis the curve fit is not performed over the whole Q range, only over low Q values where the small angle approximation is true (Fig 5.3). As a guide to picking a range the formula

$$Q_{max} * R_G \approx 1.3 \tag{5.1}$$

was employed which insures the validation of the small angle approximation.

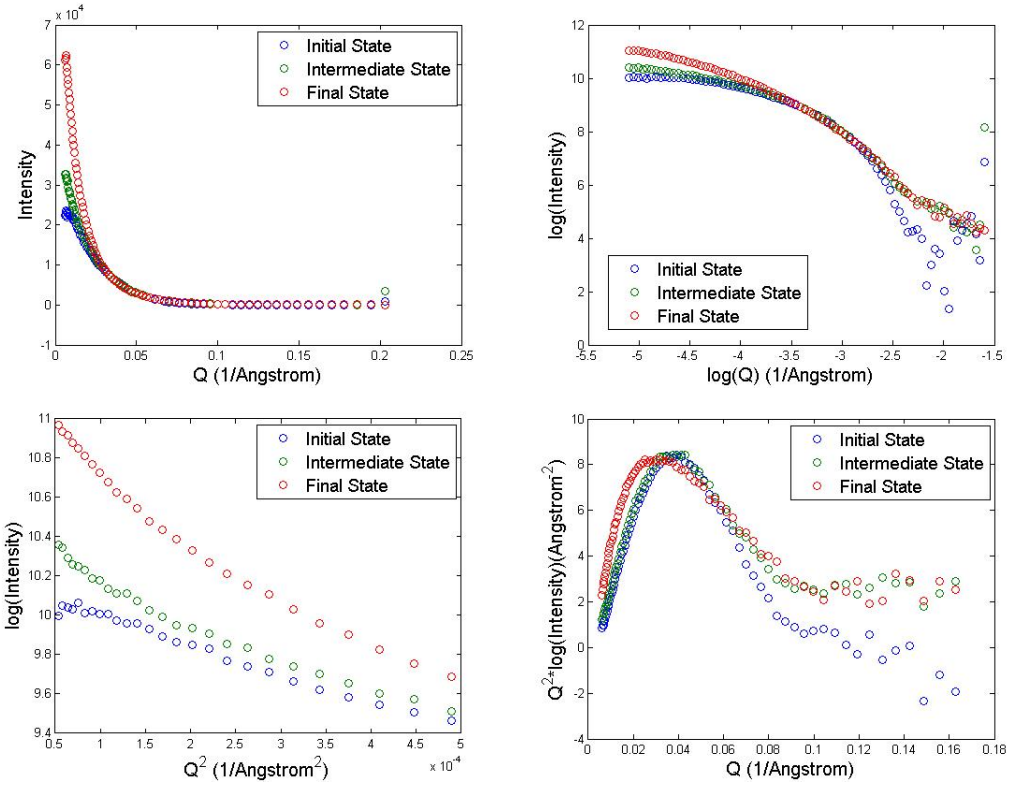


Figure 5.2: Examples of the types of plots common in SAXS analysis with initial and final protein states shown, along with an example intermediate intensity curve. Top left is an I vs Q plot. Top Right is a $Log(I)$ vs $Log(Q)$ plot. Bottom left is a Guinier Plot. Bottom right is a Kratky Plot. The initial state was taken at a temperature of 29.1 C^0 and 7 seconds, the example intermediate state at 24.9 C^0 and 183 seconds, and the final state was at 54.6 C^0 and 197 seconds.

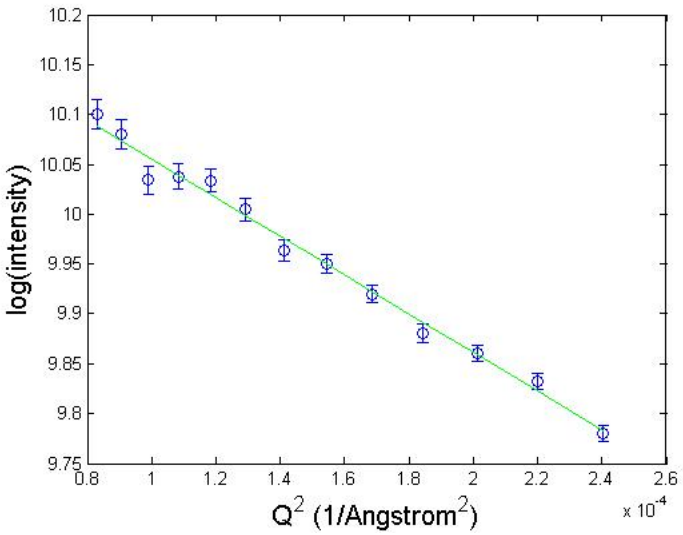


Figure 5.3: An example of a Guinier plot and a line of best fit (Slope= -1.97 $*$ 10^3 $\pm$ 71.0 $\mathring{A}^2$), at 21.5 C^0 and 69 seconds. The corresponding R_G was found to be 76.2 $\pm$ 1.4 $\mathring{A}$.

5.3 Ahhrenius and Eyring Analysis

Next values of R_1 and R_2 were determined based on the minimum and maximum calculated R_G's from the data set. R_1 and R_2 correspond to the R_G's of state 1 and state 2. They were found to be, $R_1 = 65.1$ $\mathring{A}$ and $R_2 = 101$ $\mathring{A}$. Then a plot of p_1 as a function of time was produced.(Fig. 5.4)

In general we see that as time or temperature increases the population of state one decreases or equivalently, the population of state two increases.

Again it was found necessary to remove population outliers. First all populations at time > 60 seconds were removed. This was done because for a few temperatures the populations at early times were found to be unreasonably low (≈ 0.1). The reason all, and not just the low ones, were removed is in the next step of the analysis a curve fit will be done to determine the rate coefficient, κ. Since κ is thought to be a function of time, the fit must be done over the same time range.

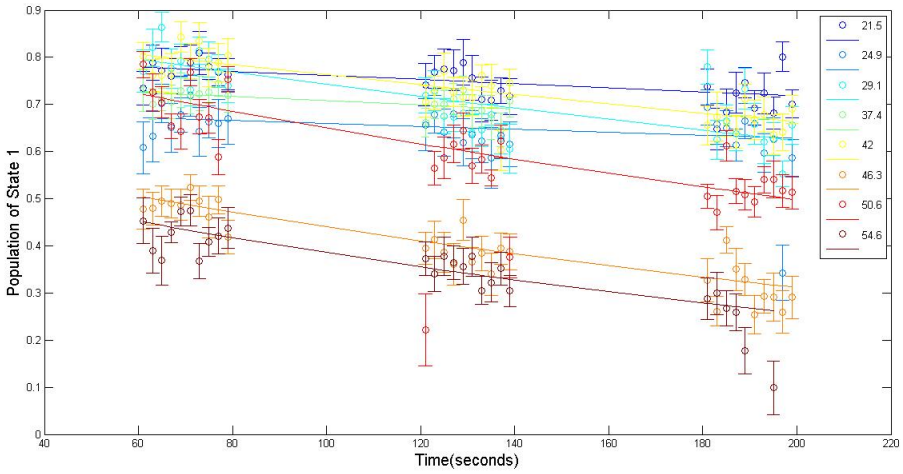


Figure 5.4: Population of state 1 as a function of time. Colors indicate different temperatures.

Next a linear curve fit of $ln(population)$ vs $time$ at each temperature was performed (Fig 5.4). The slope of this line was interpreted to be the rate coefficient, κ. Then by plotting the natural log of these rate coefficients as a function of $\frac{1}{T}$ a linear Ahhrenius plot is produced (Sec. 3.2), from which the activation energy of the transition from state one to state two is found using equation (3.23). The activation energy was measured to be 37.6 $\pm$ 14.1 kJ/mol.

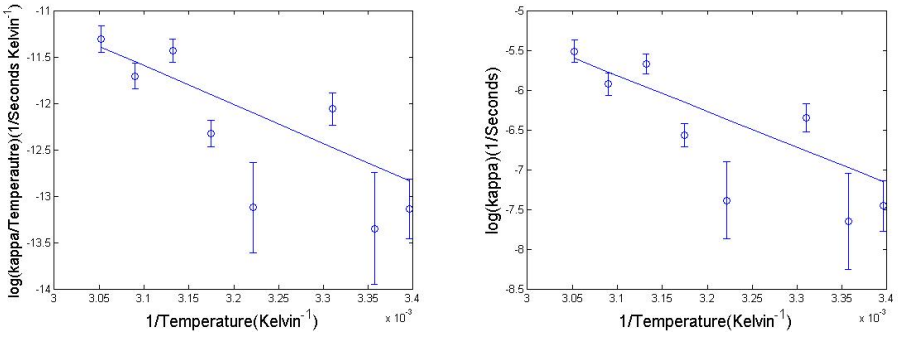


Figure 5.5: The linearized Eyring and Ahhrenius plots.

Also linear curve fit to the Eyring equation $ln(\frac{\kappa}{T})\ vs\ \frac{1}{T}$ was performed and equations (3.20) and (3.21) were used to calculate the entropy and enthalpy. The enthalpy was measured to be 35 ± 14 kJ/mol, and the entropy was -1.06 ± 0.04 kJ/(molK^{-1}). The meaning of these values will be discussed in Sec. 5.5.

In both the Ahhrenius and Eyring fits the error bar becomes larger at low temperatures. This is attributed to the rate being less at low temperatures (so the reaction is taking longer) but the time interval that the reaction is observed remains fixed at all temperatures.

5.4 Data Decomposition

The Radius of Gyration analysis employed in the previous section has been used to estimate population dynamics, from which we have determined basic thermodynamic parameters of F protein activation. We anticipate that these measurements will be confirmed by future dynamics studies using other experimental techniques such as time-resolved fluorescence. However the SAXS data is considerably richer, because in principle the entire structural history of the reaction has been recorded. In principle, each scattering curve (i.e. each experimental data point on Fig. 5.4) corresponds to a unique summation of initial, final, and perhaps other intermediate protein states. Unfortunately there is no established technique for extracting the basis scattering curves directly from these data sets.

We have chosen to employ a new technique developed for this purpose that uses constrained Singular Value Decomposition to factor the data set into a minimal set of basis functions which are forced to match particular physical criteria, such as non-negative intensities, normalized populations and other criteria followed by all SAXS I(Q) curves. (Landahl, E.C. and S.E. Rice (2013)). Although this matrix factorization is not unique, it can provide a physically realistic set of protein states and matching populations. A 3-component decomposition was done for the background-subtracted and outlier filtered time-resolved F-protein data. The pop-

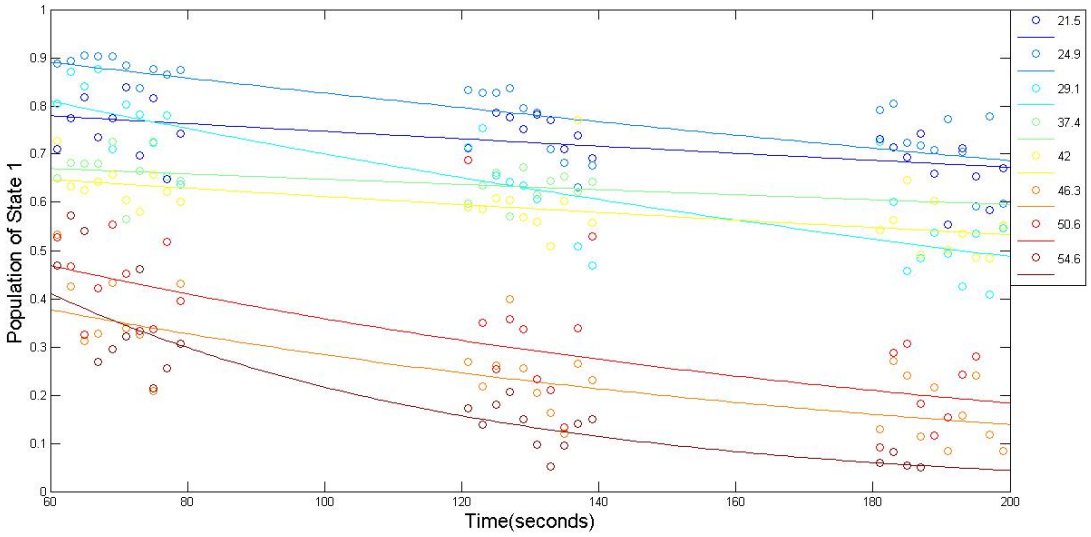


Figure 5.6: Population of state one from singular value decomposition analysis.

ulation in the first state is shown in Fig. (5.6) . Scattering curves corresponding to the first two states are shown in Fig.(5.7) , and their corresponding 3D reconstructions are shown in Fig (5.8) . Three-dimensional SAXS reconstructions were done using the software package DAMMIF (Franke, D. and Svergun, D.I. (2009)). The third state consisted of a partially disordered extended protein with a very large Rg that was generally found to contribute less than 10 % of the scattering intensity, and has been neglected from further analysis. The analysis performed on the populations found through the Radius of Gyration was then repeated on the populations found with this new technique. The activation energy was calculated to be 54 ± 14 kJ/mol, the enthalpy was found to be 51.3 ± 14.0 kJ/mol and the entropy was found to be -1.01 ±0.04 kJ/molK^{-1}.

A comparison of Figures (5.4) and (5.6) shows qualitative agreement of the population of state 1 as a function of temperature. Furthermore the thermodynamic parameters extracted agree within error.

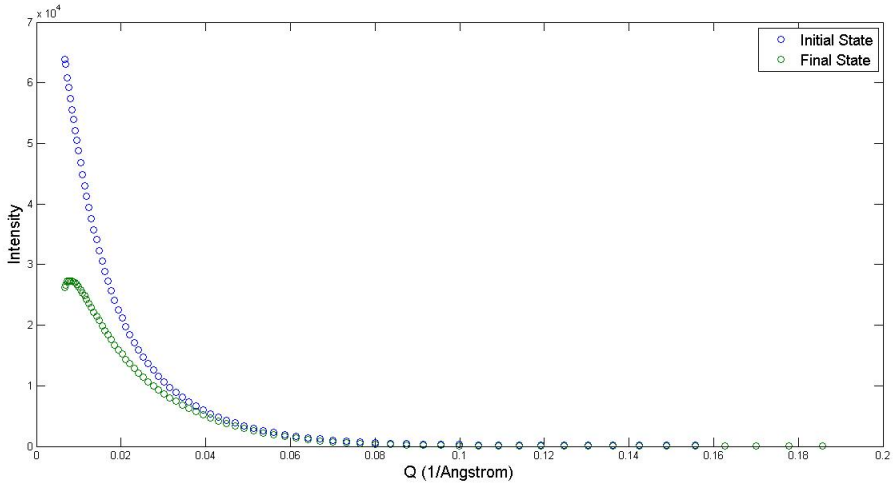


Figure 5.7: The "basis" intensities functions for the intermediate and final states generated with decomposition analysis

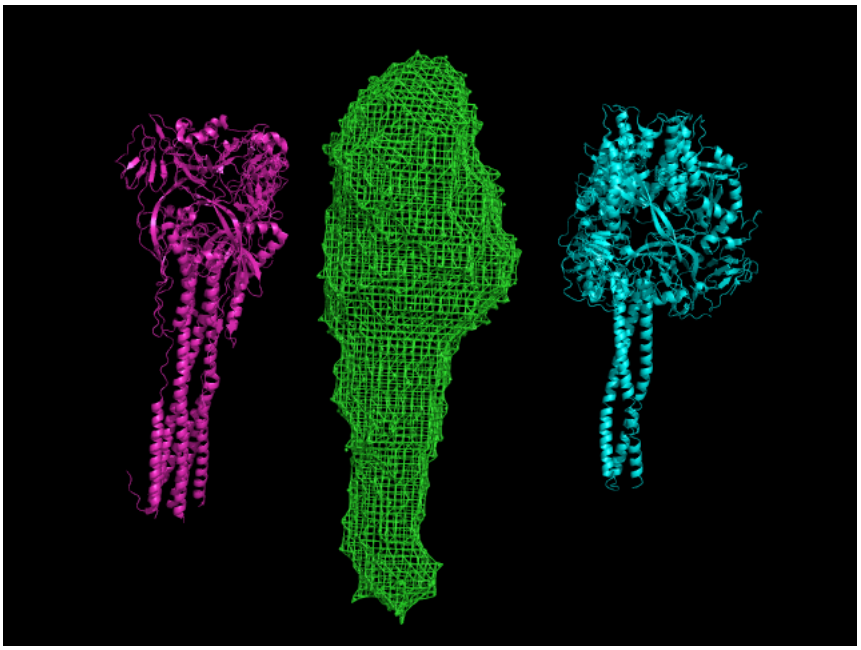


Figure 5.8: The prefusion (purple) and post fusion (blue) crystal structures compared to the SAXS reconstruction (Green)

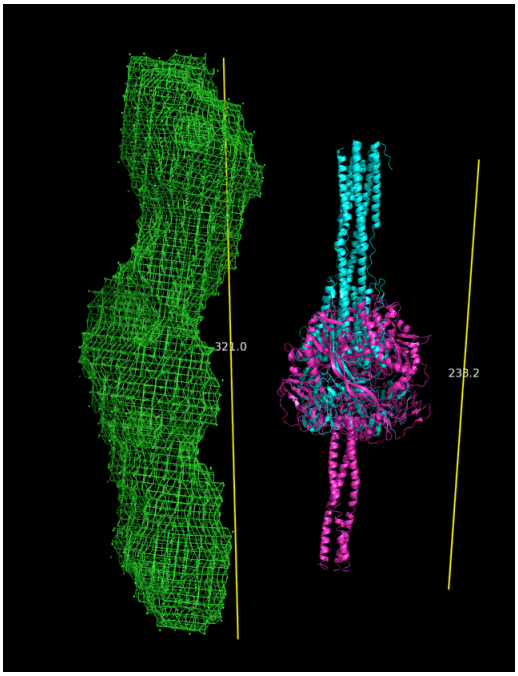


Figure 5.9: (Left) reconstructed final state found in the experiment. (Right) A Hypothetical transition state composed of the crystallized prefusion, purple, and post-fusion, blue, states.

5.5 Discussion of Results

The enthalpy and entropy found above are values associated with the rate of loss of state 1; i.e. the changes measured are with respect to the environment and not with respect to the protein. Therefore a positive enthalpy indicates an exothermic reaction and a negative entropy is an increase in multiplicity. The Gibbs free energy given by

$$\Delta G = \Delta H - T\Delta S \tag{5.2}$$

is also seen to be positive, which is interpreted to be a spontaneous reaction.

For the range of temperatures done in this experiment (295-328K) it is seen that the term, $T\Delta S$ dominates ΔH at a minimum (i.e. at T=295 K) by a factor 8.9 as seen from Eq. (5.2)

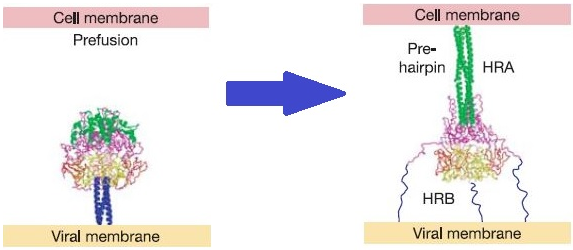


Figure 5.10: The prefusion state is transforming to some intermediate state during the infection process. The unravelling of the HRB and HRA helices implies a small enthalpic and large entropic change.

To understand why the entropy is so dominant consider the following. SAXS reconstruction from the decomposition technique of state 2 gives the green structure shown in (Fig. 5.9).The reconstruction is comparable to a hypothetical transition state comprised of pre and post fusion components taken from crystallized structures. Since it appears as though little to no post fusion protein was created from the temperature jump, the final state observed in the experiment may be this hypothetical transition state.

Now recall the proposed process of viral entry by Yin discussed in Section (1.2). After the process is initiated the HRB helices melt (Fig. 5.10). Alpha helices typically are only weekly held together by hydrogen bonds (enthalpy). Therefore the unravelling of these helices is likely to cause a gain in entropy and it is this gain in entropy which is driving the reaction forward.

CHAPTER 6

Conclusion

This thesis presented the first structural dynamic study of the fusion protein of parainfluenza virus 5. With time resolved SAXS analysis it was observed that the population of proteins was moving towards a larger radius of gyration for all temperatures and all times. This was interpreted to mean the reaction is spontaneous at all points during the experiment.

By comparison of the reconstructions shown in (Fig. 5.8) and (Fig. 5.9) it is seen that the protein is not transforming into the post-fusion state. This second state has a large radius of gyration which motivated the creation of and comparison to a hypothetical transition state with both the HRA and HRB helices unravelled, as would be consistent with Yin's model of viral entry. This transition state is consistent with the enthalpy and entropy terms found in the analysis (Sec. 5.3) and (Sec. 5.4).

Two techniques may be used as a further confirmation of the resuls here. The first would be time resolved ultra violet dichroism, which measures the difference in absorption between left and right polarized light. Helices will preferentially absorb light based on polarization. Therefore the unravelling of the helices would be observed in the diminishing difference between left and right absorption. The second technique would be time resolved florescent energy transfer. For this technique fluorophores would be attached to the helices. The light observed from these flourophores depends on the distance between them, and so can be used to determine if and when the helices unravel.

The ultimate aim in studying a virus is the prevention of its ability to infect. Since this thesis concludes that the entropic gain of the unravelling of the alpha

helices is what is driving the conformational change during the infection process; it can be suggested that a drug which can prevent the helices from unravelling would be an effective way of preventing infection.

REFERENCES

Broennimann, Ch. The Pilatus 1M Detector. Journal of Synchrotron Radiation. 2005.

Connolly SA, Leser GP, Yin HS, Jardetzky TS, Lamb RA. Refolding of a paramyxovirus F protein from prefusion to postfusion conformations observed by liposome binding and electron microscopy Proc. Natl. Acad. Sci. U. S. A. 103:1790317908

Dill. K, Molecular Driving Forces. 2nd ed. Garland Science. 2011

Franke, D. and Svergun, D.I. DAMMIF, a program for rapid ab-initio shape determination in small-angle scattering. J. Appl. Cryst., 42, 342-346

Fischetti,R. The Bio-Cat undulator beamline 18ID: a facility for biological non-crystalline diffraction and X-ray absorption spectroscopy at the Advanced Photon Source.

Garcia, A.Numerical Methods for Physics. 2nd ed. 1999

Guinier,A. X-Ray Diffraction in Crystals, Imperfect Crystals, and Amorphous Bodies

Landahl, E.C. and S.E. Rice. Model Independent Decomposition of Two-Dimensional Spectroscopic Data.(Submitted 2013)

Middleton, J. Journal of Virology. Vol. 76. No.3. 1051.

Nielsen, J. McMorrow, D. Elements of Modern X-ray Physics. 2nd ed. Wiley. 2011

Taylor,J. An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. 1996

Yin,HS. The structure of Parainfluenza virus 5 F Protein in its metastable prefusion conformation. Nature. 439

Yin,HS. Structure of the uncleaved ectodomain of the paramyxovirus (hPIV3) fusion protein. Proc.Natl.Acad.Sci.USA 102.9288-9293

6.1 Appendix of MATLAB codes

```
%Matt Andorf. program imports data(angle, intensity,error) from selected runs to qXn
%indexes runs and removes outliers.
%important note: the files in this folder for run 31 are made from the data of run 32
%updated to write outlier data to own matrices
clear all
close all
%% input runs to be processed. the nth run in 'back' must allign with the proper inde
back = [31 32 33 34 39 40 41 42 47 48 49 50 55 56 57 58 63 64 65 66 73 74 75 76 81 82
samp = [35 36 37 38 43 44 45 46 51 52 53 54 59 60 60 62 68 69 70 71 77 78 79 80 85 86
first_shot=1;      %range of shots to be imported
last_shot=10;
runs_size=size(back); %size of total runs imported
% temps and times for each shot
m=1; %for indexing
temperature = [21.5 21.5 21.5 21.5 24.9 24.9 24.9 24.9 29.1 29.1 29.1 29.1 33.3 33.3
for i = 1:length(temperature)
    for j = 1:10
        temps(m)=temperature(i);
        times(m)=(mod(i-1,4))*60 + (j-1)*2 +1;
        m = m+1;
    end
end;
clear i; clear j; clear m;
```

```
j=1;

 for k=1:runs_size(2)  %loop imports data from folder
 for i=first_shot:last_shot
  x=['run' num2str(samp(k)) '_'  num2str(i)];
%i.e. 'run31' string is created
  y=['run' num2str(back(k)) '_'  num2str(i)];
  % %%bring in data using built in function
 imp_data1=importdata(x);     %will be a 3X80 data set
 imp_data2=importdata(y);
 samp_intensity(:,j)=imp_data1(:,2);  %2nd column of data set is intensity
 samp_error(:,j)=imp_data1(:,3);      %3rd column is error
 back_intensity(:,j)=imp_data2(:,2);
 back_error(:,j)=imp_data2(:,3);
 j=j+1;
 end
 end
 bsub_intensity=samp_intensity-back_intensity;
%Background subtracted intensity
 bsub_error=sqrt(samp_error.^2+back_error.^2);
% error (summed in quadrature) for background intensity

 bsub_size=size(bsub_intensity);             %used to keep track of size of data set
 shots_removed=0;                      %used to count shots removed due to negative i

 %% remove negative intensities
k=1;
j=1;
d=0;

 while k<=bsub_size(2)

    for i=1:60 %was 75         % sets max angle to check for negative intensities.
     a=bsub_intensity(i,k);
```

```
     if a<0 %was -200     %if kth row has a negative intensity, the ith run is remove

      d=d+1;
     end
    end
      if d>0
          bsub_removed(:,j)=bsub_intensity(:,k);
%write removed data to its own matrix
          back_removed(:,j)=back_intensity(:,k);
          samp_removed(:,j)=samp_intensity(:,k);
          back_error_removed(:,j)=back_error(:,k);
          samp_error_removed(:,j)=samp_error(:,k);
          temps_removed(j)=temps(k);
          times_removed(j)=times(k);
          samp_intensity(:,k)=[];        %removes data that failed negative test.
          back_intensity(:,k)=[];
          bsub_intensity(:,k)=[];
          bsub_error(:,k)=[];
          temps(k)=[];
          times(k)=[];
         shots_removed=shots_removed +1;   %tracks amount of data removed.
     b=0;
     c=1;
     j=j+1;
     else
         b=1;
         c=0;
     end
 k=k+b;
bsub_size(2)=bsub_size(2)-c;    %update datset size to reflect removed data
d=0;
 end

 %% eliminate outliers in maximum intensity
```

```
k=1;
i=1;
avgmax=mean(bsub_intensity(3,:));   %find average intensity at a low angle
shots_removed_max=0;                 %tracks how many runs are removed
while k<=bsub_size(2)
    a=max(bsub_intensity(:,k));
    if a> 2*avgmax % was 1.5 times
        standard_rat(i)=(a-avgmax)/std(bsub_intensity(3,:));
        bsub_removed(:,j)=bsub_intensity(:,k);
         back_removed(:,j)=back_intensity(:,k);
         samp_removed(:,j)=samp_intensity(:,k);
         back_error_removed(:,j)=back_error(:,k);
         samp_error_removed(:,j)=samp_error(:,k);
         temps_removed(j)=temps(k);
         times_removed(j)=times(k);
        bsub_intensity(:,k)=[];
        bsub_error(:,k)=[];
        temps(k)=[];
        times(k)=[];
        b=0;
        c=1;
        shots_removed_max=shots_removed_max+1;
        i=i+1;
    else
        b=1;
        c=0;
    end
    k=k+b;

bsub_size(2)=bsub_size(2)-c;
end
%% Replace shots with zero error.

 k=1;
```

```
 for k=1:bsub_size(2)  %some shots were missing error so surogate error is used.
     a=bsub_error(1,k);
     if a==0 % was 1.5 times
         bsub_error(:,k)=bsub_error(:,1); %replace the kth column error, with the 1st
        end
end
angle=imp_data1(:,1);    %creates 1x80 scattering vector.


%Matt Andorf
%% Performs A gunier anaylsis to find Rg
%Turns Rg's into populations f_1 and f_2.
% curve fitting routine from LinFit_demo.m by Eric Landahl
%modified to include all data
%MUST RUN readremoveindex_3.m FIRST


minpt=10;  %minimum q value in analysis
maxpt=22;  %maximum q value
xg=bsub_intensity(minpt:maxpt,:); %extract intensity
eg=bsub_error(minpt:maxpt,:);
qq=angle(minpt:maxpt);




%%begin Curve fitting routine
for k=1:bsub_size(2)
XX = qq.^2';      %scattering vector
YY = log(xg(:,k))'; %log of intensity
Y=YY';   % Convert y-data to a column vector
SS = (eg(:,k)'./xg(:,k)');  % from error propagation
```

```
VV = SS.^2;  % standard deviation at each point
Var=sum(SS.^2);  %total variance
V = diag(VV);  % for uncorrelated errors, the variance matrix is a diagonal matrix
W=inv(V)*Var;  % data weights are 1/variance
[nu numpts]=size(XX);  %nu is not used, numpts is number of data points
% We have the function Y(X) = m1 + m2*X
numparams=2;  % two parameters m1 and m2 for linear problem
i=1:numpts;
j=1:numparams;
X(i,1)=1;  % derivitive of function wrt first fitting parameter
X(i,2)=XX(i);  % derivitive of function wrt second fitting parameter
P=inv(X' * W * X);  % Intermediate matrix calculated for least squares fit and errors
M(:,k)=P* X' * W * Y;  % results of least square fit
Ycalc(:,k)=X*M(:,k) ;  % Use least square fit parameters to calculate the fit line
E=Ycalc(:,k)-Y ;  % Difference between fit and data
S=(E' * W * E);  % Sum of squared residuals
sigmas_res(:,k)=(( S *diag(P))/(numpts-numparams)).^(0.5); % Error in parameters due
sigmas_w(:,k)  =((Var*diag(P))/(numpts-numparams)).^(0.5);
% Error in parameters due to error bars
sigmas_tot(:,k)=sigmas_res(:,k) + sigmas_w(:,k);
% total error in each fit parameter
end
%%find Rg from slope

Rg=sqrt(-3*M(2,:));   %formula for Rg (Angstroms)
Rg_error=3*(0.5)*sigmas_tot(2,:).*((-3*M(2,:)).^(-1/2));
%error in Rg  (angstroms)
Rg_check=angle(maxpt)*Rg; %values should be around 1.3. Used to determine a good Q ra
Rg_avg=sum(Rg_check)/length(Rg_check); %average value should be around 1.3.


%%find populations from Rg
[roo coll]=find(max(Rg)==Rg,100);  %coll is the column with the maxiumum Rg
R2=Rg(coll);  %R2 is the maximum Rg
```

```
[rooo colll]=find(min(Rg)==Rg,100); %colll is the column with the minimum Rg
R1=Rg(colll);     %picked since it is one of the smaller Rg's
f_1=(Rg.*Rg-R2*R2)/(R1*R1-R2*R2);   %equation for population 1.
f_1_error=abs(2*Rg.*Rg_error/(R1*R1-R2*R2));  %error in population 1
f_2=1-f_1;       %population of 2 (never used)


%Matt Andorf.
% plot population values as a function for time for a specific temperature
%linear curve fits log of populations vs time
% Performs ahhrenius and Eyring analysis based on curve fits of populations
%run readremoveindex_3.m AND gunier_curvefit.m first

%%
clear M_f                  %clear variables from previous runs of the program
clear sigmas_tot_f
clear sigmas_w_f
clear sigmas_res_f
clear xxx
clear temp
k=1;
kk=1;
temp=[21.5,24.9,29.1,37.4,42,46.3,50.6,54.6];
%temperature sets to be analized (Celsius)
colrr=colormap(jet(length(temp)));    %used for plotting
figure(1);%clf;hold on
 for kk=1:length(temp)
     clear Rg_temp     %clear variables for each itteration
clear Rg_time
clear f_1_temp
clear f_1_temp_error
clear XX_time
clear YY_f
clear Y_f
```

```
clear X_f
clear W_f
clear V_f
clear V_f
clear nu_f
clear numpts_f
clear Ycalc_f
clear E_f
clear S_f
clear figure(1)

[ro col]=find(temp(kk)==temps,100);  %col is a vector of column indices containing th
for k=1:length(col)
Rg_time(k)=times(col(k));  %put times of specified temperature in a vecor
f_l_temp(k)=f_1(col(k));%put populations of specified temperature in a vector
f_l_temp_error(k)=f_1_error(col(k)); %put error of population from specified temperat
end


 %% remove populations greater than 1. (This occurs since R1 is not neccesarily the s
ii=1;
poplength=length(f_l_temp);
while ii<length(f_l_temp)
    a=f_l_temp(ii);
     if a>1

     Rg_time(ii)=[];
     f_l_temp(ii)=[];
    f_l_temp_error(ii)=[];
     b=0;
     c=1;
     else
         b=1;
         c=0;
```

```
     end
 ii=ii+b;
 poplength=poplength-c;
    end
%% remove outliers (small populations)
ii=1;
while ii<=length(f_l_temp)
     pop_avg=mean(f_l_temp(:)); %find average population at specific temperature
     a=f_l_temp(ii)
     if a<0.1*pop_avg %if the iith population is less then 0.1 of the average it is

     Rg_time(ii)=[]; %updata data to reflect removed population
     f_l_temp(ii)=[];
     f_l_temp_error(ii)=[];
     b=0;
     c=1;
     else
         b=1;
         c=0;
     end
 ii=ii+b;


end

% remove populations at time<60.
ii=1;
poplength=length(f_l_temp);
while ii<length(f_l_temp)
    a=Rg_time(ii);
     if a<60

     Rg_time(ii)=[];
     f_l_temp(ii)=[];
```

```
    f_l_temp_error(ii)=[];
     b=0;
     c=1;
     else
         b=1;
         c=0;
     end
 ii=ii+b;
 poplength=poplength-c;
    end



%%begin Curve fitting routine
XX_time = Rg_time;
YY_f =log(f_l_temp);
Y_f=YY_f';   % Convert y-data to a column vector
SS_f =  (f_l_temp_error./f_l_temp);  %
VV_f = SS_f.^2;  % standard deviation at each point
Var_f=sum(SS_f.^2);  %total variance
V_f = diag(VV_f);  % for uncorrelated errors, the variance matrix is a diagonal matri
W_f=inv(V_f)*Var_f;  % data weights are 1/variance
[nu_f numpts_f]=size(XX_time);  %nu is not used, numpts is number of data points
% We have the function Y(X) = m1 + m2*X
numparams_f=2;  % two parameters m1 and m2 for linear problem
i=1:numpts_f;
j=1:numparams_f;
X_f(i,1)=1;  % derivitive of function wrt first fitting parameter
X_f(i,2)=XX_time(i);  % derivitive of function wrt second fitting parameter
P_f=inv(X_f' * W_f * X_f);  % Intermediate matrix calculated for least squares fit ar
M_f(:,kk)=P_f* X_f' * W_f * Y_f;  % results of least square fit
Ycalc_f=X_f*M_f(:,kk) ;  % Use least square fit parameters to calculate the fit line
E_f=Ycalc_f-Y_f ;  % Difference between fit and data
S_f=(E_f' * W_f * E_f);  % Sum of squared residuals
```

```
sigmas_res_f(:,kk)=(( S_f *diag(P_f))/(numpts_f-numparams_f)).^(0.5); % Error in para
sigmas_w_f(:,kk)  =((Var_f*diag(P_f))/(numpts_f-numparams_f)).^(0.5);
% Error in parameters due to error bars
sigmas_tot_f(:,kk)=sigmas_res_f(:,kk) + sigmas_w_f(:,kk);
% total error in each fit parameter
%% Generate plots
longtime=0:1:1500; %extend curve fit.
errorbar(Rg_time, f_l_temp,f_l_temp_error,'o', 'color',colrr(kk,:)) %plot data with e
   hold on
plot(longtime, exp(M_f(2,kk)*longtime+M_f(1,kk)),'color',colrr(kk,:))
%plot curve fit
legend('21.5','','24.9','','29.1','','37.4','','42','','46.3','','50.6','','54.6','')
%generate legend
xhandle=xlabel('Time(seconds)'); %label axis
yhandle=ylabel('Population of State 1');
set(xhandle,'Fontsize',15)  %change font size
set(yhandle,'Fontsize',15)
 end
rate=abs(M_f(2,:));  %the slope of log(population) vs time is the rate
rate_error=sigmas_tot_f(2,:);  %slope error from both error bars and residue
temp_act=temp(1:8)+273;   %Convert temperatures to Kelvin

%% Curve fit log of rate vs 1/temperature to for ahhrenius plot
XX_temp = 1./temp_act;
YY_rate =log(rate);
Y_rate=YY_rate';   % Convert y-data to a column vector
SS_rate = (rate_error./rate);  % Assume errors are uncorrelated.
VV_rate = SS_rate.^2;  % standard deviation at each point
Var_rate=sum(SS_rate.^2);  %total variance
V_rate = diag(VV_rate);  % for uncorrelated errors, the variance matrix is a diagonal
W_rate=inv(V_rate)*Var_rate;  % data weights are 1/variance
[nu_rate numpts_rate]=size(XX_temp);  %nu is not used, numpts is number of data point
% We have the function Y(X) = m1 + m2*X
numparams_rate=2;  % two parameters m1 and m2 for linear problem
```

```
i=1:numpts_rate;
j=1:numparams_rate;
X_rate(i,1)=1;  % derivitive of function wrt first fitting parameter
X_rate(i,2)=XX_temp(i);  % derivitive of function wrt second fitting parameter
P_rate=inv(X_rate' * W_rate * X_rate);  % Intermediate matrix calculated for least sq
M_rate=P_rate* X_rate' * W_rate * Y_rate;  % results of least square fit
Ycalc_rate=X_rate*M_rate ;  % Use least square fit parameters to calculate the fit li
E_rate=Ycalc_rate-Y_rate ;  % Difference between fit and data
S_rate=(E_rate' * W_rate * E_rate);  % Sum of squared residuals
sigmas_res_rate=(( S_rate *diag(P_rate))/(numpts_rate-numparams_rate)).^(0.5); % Erro
sigmas_w_rate =((Var_rate*diag(P_rate))/(numpts_rate-numparams_rate)).^(0.5);
% Error in parameters due to error bars
sigmas_tot_rate=sigmas_res_rate + sigmas_w_rate;
% total error in each fit parameter

%%find activation energy

Activation_Energy= -8.314*M_rate(2)/1000 %in kJ/mole
Activation_Error=-8.314*sigmas_tot_rate(2)/1000 %kJ/mole


%Eyring Analysis
 kot=rate./(temp+273);   %write a vector of k (rate constant)/Temperature
 kot_error=rate_error./(temp+273);  %error kot
 kb=1.380*10^23;       %boltsmann constant joules/Kelvin
 h=6.626*10^-34;       %Plank consant joule*seconds
 R=8.314;            %ideal gas constant

 %% linear fit Eyring Equation
 XX_temp = 1./temp_act;
YY_kot =log(kot);
Y_kot=YY_kot';   % Convert y-data to a column vector
SS_kot = kot_error./kot;  % Assume errors are uncorrelated.
VV_kot = SS_kot.^2;  % standard deviation at each point
```

```
Var_kot=sum(SS_kot.^2);  %total variance
V_kot = diag(VV_kot);  % for uncorrelated errors, the variance matrix is a diagonal m
W_kot=inv(V_kot)*Var_kot;  % data weights are 1/variance
[nu_kot numpts_kot]=size(XX_temp);  %nu is not used, numpts is number of data points
% We have the function Y(X) = m1 + m2*X
numparams_kot=2;  % two parameters m1 and m2 for linear problem
i=1:numpts_kot;
j=1:numparams_kot;
X_kot(i,1)=1;  % derivitive of function wrt first fitting parameter
X_kot(i,2)=XX_temp(i);  % derivitive of function wrt second fitting parameter
P_kot=inv(X_kot' * W_kot * X_kot);  % Intermediate matrix calculated for least square
M_kot=P_kot* X_kot' * W_kot * Y_kot;  % results of least square fit
Ycalc_kot=X_kot*M_kot ;  % Use least square fit parameters to calculate the fit line
E_kot=Ycalc_kot-Y_kot ;  % Difference between fit and data
S_kot=(E_kot' * W_kot * E_kot);  % Sum of squared residuals
sigmas_res_kot=(( S_kot *diag(P_kot))/(numpts_kot-numparams_kot)).^(0.5); % Error in
sigmas_w_kot =((Var_kot*diag(P_kot))/(numpts_kot-numparams_kot)).^(0.5);
% Error in parameters due to error bars
sigmas_tot_kot=sigmas_res_kot + sigmas_w_kot;
% total error in each fit parameter

Enthalpy=-R*M_kot(2)/1000       %in kJ/mole
Enthalpy_error=-R*sigmas_tot_kot(2)/1000  %enthalpy error kJ/mole

Entropy=R*(M_kot(1)-log(kb/h))/1000  %kJ/(Mole*K)
Entropy_error=R*(sigmas_tot_kot(1))/1000 %kJ/(mole*K)




%% create Eyring and Ahhrenius plots
figure(2)
errorbar(1./temp_act,log(rate),SS_rate,'o') %plot 1/temp vs log of rate and error (te
 hold on
```

```
 plot(1./temp_act,Ycalc_rate)     %plot Ahhrenius curve fit
 xhandle=xlabel('1/Temperature(Kelvin^-^1)');  %label axis
 yhandle=ylabel('log(kappa)(1/Seconds)');
 set(xhandle,'Fontsize',15)    %change font size
 set(yhandle,'Fontsize',15)


 figure(3)
 errorbar(1./temp_act,log(abs(kot)),SS_kot,'o')
% 1/temperature vs log(rate/temperature) (temp in Kelvin)
 hold on
 plot(1./temp_act,Ycalc_kot)     %plot curve fit
 xhandle=xlabel('1/Temperature(Kelvin^-^1)');   %label axis
 yhandle=ylabel('log(kappa/Temperautre)(1/Seconds Kelvin^-^1)');
 set(xhandle,'Fontsize',15)  %change font size
set(yhandle,'Fontsize',13)


%creates various plots
%run readremoveindex_3.m first
mn=5;   %min and max values for plots
mx=30;

back_plot=back_intensity(:,5);   %the 5th column from the back and sample intensities
samp_plot=samp_intensity(:,5);    %as an example

figure(1)   %this is a plot of a sample and background intensity vs Q of a removed se
plot(angle(mn:mx),back_removed(mn:mx,36),'x')
hold on
plot(angle(mn:mx),samp_removed(mn:mx,36),'o')
  hold on
  yhandle=ylabel('Intensity')
  xhandle=xlabel('Q (1/Angstrom)')
 lhandle= legend('background','sample','background subtracted')
```

```
   set(xhandle,'Fontsize',15)
 set(yhandle,'Fontsize',15)
 set(lhandle,'Fontsize',15)

  figure(2)     %creates plot of a background and sample intensity vs Q of a "normal'
  plot(angle(mn:mx),back_plot(mn:mx),'x')
hold on
plot(angle(mn:mx),samp_plot(mn:mx),'o')
  hold on

  yhandle=ylabel('Intensity')
  xhandle=xlabel('Q (1/Angstrom)')
   set(xhandle,'Fontsize',15)
 set(yhandle,'Fontsize',15)
  lhandle=legend('background','sample','background subtracted')
  set(lhandle,'Fontsize',15)

  figure(3)      % plots a guinier plot (variables from guinier_curvefit.m
   errorbar(XX,log(xg(:,5)),(eg(:,5)./xg(:,5)),'o')
   hold on
   plot(XX,M(2,5)*XX+M(1,5),'g')


xhandle=xlabel('Q^2 (1/Angstrom^2)')
yhandle=ylabel('log(intensity)')
set(xhandle,'Fontsize',15)
 set(yhandle,'Fontsize',15)
% plot guinier, log log, raw and kratky of initial, intermediate and final states
beg=bsub_intensity(:,10);  % intensity produced a small Rg
int=bsub_intensity(:,55);  %intensity of a minimum Rg
fin=bsub_intensity(:,coll);  %intensity of the maximum Rg
beg_temp=temps(65) %temp of beg (celsius)
beg_time=times(65) %time of beg (seconds)
int_temp=temps(55) %temp of int (celsius)
```

```
int_time=times(55)  %time of int (seconds)
fin_temp=temps(coll) %temp of fin (celsius)
fin_times=times(coll) %time of fin (seconds)

figure(4)  %generates plot of Intensity Vs Q for each state
plot(angle,beg,'o',angle,int,'o',angle,fin,'o');
hold on
plot(angle,int,'x');
hold on
plot(angle,fin,'d');
xhandle=xlabel('Q (1/Angstrom)')
yhandle=ylabel('Intensity')

set(xhandle,'Fontsize',15)
set(yhandle,'Fontsize',15)

lhandle=legend('Initial State','Intermediate State','Final State')
set(lhandle,'Fontsize',14)


figure(5)   %generates log log plot for each state
plot(log(angle),log(beg),'o',log(angle),log(int),'o',log(angle),log(fin),'o');
hold on
plot(log(angle),log(int),'b');
hold on
plot(log(angle),log(fin),'r');
xhandle=xlabel('log(Q) (1/Angstrom)')
yhandle=ylabel('log(Intensity)')
set(xhandle,'Fontsize',15)
set(yhandle,'Fontsize',15)
lhandle=legend('Initial State','Intermediate State','Final State')
set(lhandle,'Fontsize',14)

figure(6)    %generates Guinier plot for each state
```

```
plot(angle(mn:mx).^2,log(beg(mn:mx)),'o',angle(mn:mx).^2,log(int(mn:mx)),'o',angle(mn
hold on
plot(angle(mn:mx).^2,log(int(mn:mx)),'b');
hold on
plot(angle(mn:mx).^2,log(fin(mn:mx)),'r');
xhandle=xlabel('Q^2 (1/Angstrom^2)')
yhandle=ylabel('log(Intensity)')
set(xhandle,'Fontsize',15)
set(yhandle,'Fontsize',15)
lhandle=legend('Initial State','Intermediate State','Final State')
set(lhandle,'Fontsize',14)


figure(7)  %generates Kratky plot for each state
kn=1;
kx=75;
plot(angle(kn:kx),beg(kn:kx).*angle(kn:kx).^2,'o',(angle(kn:kx)),int(kn:kx).*angle(kn
hold on
plot(angle(kn:kx),int(kn:kx).*angle(kn:kx).^2,'b');
hold on
plot(angle(kn:kx),fin(kn:kx).*angle(kn:kx).^2,'r');
xhandle=xlabel('Q (1/Angstrom)')
yhandle=ylabel('Q^2*log(Intensity)(Angstrom^-^2)')
set(xhandle,'Fontsize',15)
set(yhandle,'Fontsize',15)
lhandle=legend('Initial State','Intermediate State','Final State')
set(lhandle,'Fontsize',14)
```